Predicting dark matter halo formation in N-body simulations with deep regression networks

Bernardini, M ; Mayer, L ; Reed, D ; Feldmann, R

Abstract: Dark matter haloes play a fundamental role in cosmological structure formation. The most common approach to model their assembly mechanisms is through N-body simulations. In this work, we present an innovative pathway to predict dark matter halo formation from the initial density field using a Deep Learning algorithm. We implement and train a Deep Convolutional Neural Network to solve the task of retrieving Lagrangian patches from which dark matter haloes will condense. The volumetric multilabel classification task is turned into a regression problem by means of the Euclidean distance transformation. The network is complemented by an adaptive version of the watershed algorithm to form the entire protohalo identification pipeline. We show that splitting the segmentation problem into two distinct subtasks allows for training smaller and faster networks, while the predictive power of the pipeline remains the same. The model is trained on synthetic data derived from a single full N-body simulation and achieves deviations of 10 per cent when reconstructing the dark matter halo mass function at z = 0. This approach represents a promising framework for learning highly non-linear relations in the primordial density field. As a practical application, our method can be used to produce mock dark matter halo catalogues directly from the initial conditions of N-body simulations.

DOI: https://doi.org/10.1093/mnras/staa1911

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **496,** 5116–5125 (2020) doi:10.1093/mnras/staa1911
Advance Access publication 2020 July 07

Predicting dark matter halo formation in *N*-body simulations with deep regression networks

M. Bernardini,* L. Mayer, D. Reed and R. Feldmann

Center for Theoretical Astrophysics and Cosmology, Institute for Computational Science, University of Zurich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland

Accepted 2020 June 25. Received 2020 June 19; in original form 2019 December 11

ABSTRACT

Dark matter haloes play a fundamental role in cosmological structure formation. The most common approach to model their assembly mechanisms is through *N*-body simulations. In this work, we present an innovative pathway to predict dark matter halo formation from the initial density field using a Deep Learning algorithm. We implement and train a Deep Convolutional Neural Network to solve the task of retrieving Lagrangian patches from which dark matter haloes will condense. The volumetric multilabel classification task is turned into a regression problem by means of the Euclidean distance transformation. The network is complemented by an adaptive version of the watershed algorithm to form the entire protohalo identification pipeline. We show that splitting the segmentation problem into two distinct subtasks allows for training smaller and faster networks, while the predictive power of the pipeline remains the same. The model is trained on synthetic data derived from a single full *N*-body simulation and achieves deviations of $\sim$10 per cent when reconstructing the dark matter halo mass function at $z = 0$. This approach represents a promising framework for learning highly non-linear relations in the primordial density field. As a practical application, our method can be used to produce mock dark matter halo catalogues directly from the initial conditions of *N*-body simulations.

Key words: methods: numerical – galaxies: haloes – large-scale structure of Universe – dark matter.

1 INTRODUCTION

The fundamental information source of gravitational dynamics is the cosmic density field described by its non-linear evolution. Cosmological *N*-body simulations of cold dark matter (CDM) show how initially overdense regions collapse through the competition of cosmic expansion and gravity to form virialized structures called dark matter haloes. These haloes form the building blocks of large-scale structure as they define the landscape of potential wells in which baryonic matter flows to form galaxies, groups, and clusters of galaxies (e.g. Guo et al. 2010; Wechsler & Tinker 2018). The structure and formation of dark matter haloes is thus an important mechanism to understand when building a complete model of galaxy formation and evolution as intrinsic galaxy properties depend on the host dark matter halo mass and morphology (Wechsler & Tinker 2018; Feldmann, Faucher-Giguère & Kereš 2019).

The source of structure formation lies in the primordial perturbations seeded in the matter field of the early Universe, where initially small density peaks grow linearly through mass accretion and mergers of smaller structures. As the evolution of the matter density field progresses into the non-linear regime, the continuing gravitational accretion starts to form distinct virialized objects, that strongly affect their corresponding environments and substructures. As individual systems transition into the fully non-linear regime, the assembly history of dark matter haloes becomes generally difficult as complicated gravitational affects like merging or tidal stripping of larger structures can occur. In this regime, *N*-body simulations provide the only reliable tool for accurately describing structure formation processes.

The development of analytical approximations for describing where and how structure forms in an initial random field has opened the possibility to understand the main physical aspects that drive halo assembly. The fundamental analytical work by Press & Schechter (1974) suggests that dark matter collapse occurs once the spherically smoothed linear density field exceeds a cosmology-dependent threshold value. This idea is complemented by the excursion set formalism by Bond et al. (1991) which describes gravitational collapse as a statistical framework based on density trajectories on varying scales. This framework was further developed by relaxing the assumption of spherical collapse and calibrating with numerical simulations. These semi-analytical schemes have been shown to reproduce halo statistics with acceptable error margins providing a fast method for sampling mock dark matter halo catalogues (Sheth, Mo & Tormen 2001; Reed et al. 2003).

In recent time, a wide variety of machine learning techniques have been investigated to solve tasks related to non-linear structure formation (e.g. Agarwal, Davé & Bassett 2017; Lucie-Smith et al. 2018; Mathuriya et al. 2018; Berger & Stein 2019; Aragon-Calvo 2019; He et al. 2019; Lucie-Smith, Peiris & Pontzen 2019; Zhang et al. 2019; Kodi Ramanah, Charnock & Lavaux 2019; Charnock et al. 2020; Ntampaka et al. 2020). Generally, the approach of examining fully non-linear collapse dynamics by running *N*-body simulations is computationally expensive. Incorporating machine learning is

* E-mail: mauro.bernardini@uzh.ch

therefore a natural step, since these algorithms are very adaptive to a large variety of problems and arrive at predictions comparatively fast.

Convolutional Neural Networks (CNNs) form a branch of Deep Learning models that have drastically changed computer vision problems (Krizhevsky, Sutskever & Hinton 2012). As many other deep learning architectures, CNNs use the compositions of non-linear functions to model complex dependencies between input features and target variables. An interesting subsection of CNNs deals with the task of image segmentation, where networks are trained to identify and retrieve structures in input images. Through the combination of layered non-linear activation functions, CNNs learn derivatives in the input space to isolate objects of interest. Networks trained on segmentation tasks have a large variety of applications that range from identification of every day objects (e.g. Liu, Rabinovich & Berg 2015; Noh, Hong & Han 2015) up to nuclei segmentation in biological images of cancer tissue (e.g. Milletari, Navab & Ahmadi 2016; Isensee et al. 2018; Naylor et al. 2019).

In this work, we present an application of machine learning based on a CNN to identify Lagrangian patches (termed protohaloes) in the primordial density field, from which dark matter haloes condense. We formulate the problem in such a way that the pipeline can be used to produce mock dark matter halo catalogues directly from the initial conditions (IC) of *N*-body simulations. We design the classification task of distinguishing between collapsing and background regions as a regression problem in the distance space. This alternative technique was first proposed by Naylor et al. (2019) and shows great success for images with overlapping regions. The precision of the entire pipeline is measured by comparing the reconstructed dark matter halo mass function to its corresponding ground truth.

Since identifying collapsing regions in the primordial density field is a highly non-trivial task, the success of training a machine learning model strongly depends on how this mapping is formulated numerically. Lucie-Smith et al. (2018) trained a Random Forest on data retrieved on a particle by particle basis to predict whether or not a particle will be a member of a dark matter halo at $z = 0$. They successfully showed that the primordial density field contains enough information to predict halo membership as the precision of their algorithm outperformed analytical frameworks like the Extended Press–Schechter formalism. They also showed that the density contrast is an information carrier to predict the final distance of particles from their corresponding halo density peaks.

Berger & Stein (2019) presented a powerful binary classification network (that inspired our own implementation) and paired it with an algorithm to extract halo regions from the reconstructed probability map. Compared to the particle information in Lucie-Smith et al. (2018), their neural network operates upon gridded density contrast information. Regarding halo formation, this approach has the advantage that the network learns to identify the important features in the input map and derivatives thereof itself, rather than being presented with a fixed feature vector as in Lucie-Smith et al. (2018). This allows the network to scan the exact shape of the local neighbourhood around density peaks on different scales, in order to assess the underlying collapse dynamics. The downside of formulating halo membership in a binary approach is, that retrieving individual haloes strongly depends on border pixels having smaller probability values then centrally located cells, in order to minimize the undesired effect of haloes artificially clumping together. This indicates that the success of this method relies on the imperfection of the neural network prediction regarding the reconstructed probability map as halo borders are identified by varying probability thresholds.

Inspired by this approach, we seek to formulate a mapping where the network is trained on a target that incorporates the information of protohalo borders by construction. We deliberately focus on retrieving halo information from the ICs rather than from the $z = 0$ snapshot for two very important reasons. First, the central location of haloes changes over time through gravitational interactions as they gradually divert from the initial locations of the primordial density peaks. Second, the final halo sizes at $z = 0$ in full *N*-body simulations are small compared to the box size and thus span only a very small subsection of the simulation volume, making the target field sparsely populated. Training on sparsely populated target fields of the exact halo position and mass is a very difficult task as has been pointed out by Zhang et al. (2019, although in a slightly different application). For these reasons we choose to design the network mapping where input and target both display information at the ICs. We show that by introducing gradient fields in the target map, and in particular the distance information, one can train the network to predict membership of central pixels more accurately than border cells. This gives us the advantage of well-defined borders in the target map, as information of individual objects is retained.

This paper is structured as follows. In Section 2, we discuss the methods we used to construct the training samples for the supervised regression problem from cosmological *N*-body simulations. Then, we outline the mapping the network is trained on and in Section 3, we present our network implementation and training strategy in great detail. In Section 4, we describe an adaptive algorithm to reconstruct the segmentation map from the network output, in order to predict the dark matter halo distribution for a single simulation box. We assess the network precision by comparing the predicted and true halo statistics in form of the halo mass function at $z = 0$. We furthermore discuss distinct strengths and weaknesses of the network at its current state when confronted with specific cases of density configurations. We propose future ideas for further improvements and conclude in Section 5.

2 DEEP LEARNING FOR STRUCTURE IDENTIFICATION

In this section, we present the details of our implemented pipeline in the following sequence:

(i) We present the numerical details of the *N*-body simulations used in this work as well as the identification algorithm of dark matter haloes at $z = 0$.

(ii) We introduce the concept of a protohalo at the ICs, how these objects are retrieved from the simulation and how they are used in the prediction algorithm.

(iii) We describe the numerical concept of the distance transformation as a metric in the context of protohalo boundaries and how this information allows to construct the data samples for the network mapping.

2.1 Cosmological *N*-body simulations

We adopt two *N*-body simulations of the same ΛCDM cosmology ($\Omega_m = 0.279$, $\Omega_\Lambda = 0.721$) produced by `pkdgrav` (Stadel et al. 2000), one for training and one for testing the algorithm. The comoving box size is $100\,h^{-1}$Mpc with a particle resolution of 512^3, where each individual dark matter particle carries a physical mass of 8.84×10^8 $M_\odot$. We use the `HOP` halo finder by Eisenstein & Hut (1998) to retrieve dark matter halo catalogues from the simulations. This group finding algorithms first associates a density estimate for all particles. Each particle is then iteratively linked to its densest nearest neighbour until the algorithm reaches a particle which is its

own densest neighbour. The collection of particles that are traced to the same densest particle forms a single group. In this way, there are no constraints on the morphology of individual haloes and since this method does not rely on a linking length, the undesired effect of bridging discrete haloes together is avoided entirely. We note that the neural network mapping is unaltered when choosing a different halo finder as the network learns the concept of a protohalo from the data samples itself.

As outlined in Section 2.2, the algorithm depends on the prerequisite that the individual borders of Lagrangian patches in the ICs be sufficiently resolved. For this manner, we choose a lower halo mass threshold of $4 \times 10^{12}\,\mathrm{M}_{\odot}$ (corresponding to 4525 particles) as smaller haloes clustering around larger structures make the borders on the grid not sufficiently resolved for the underlying task. Running the halo finder over the corresponding $z = 0$ snapshots results in catalogues of 6D phase-space information for each halo. The training simulation contains 1418 dark matter haloes ranging up to $\sim 10^{15}\,\mathrm{M}_{\odot}$ where the test simulation contains 1630 thereof.

2.2 Formulating the network mapping

The primordial density field is the source of input information from which the network learns to retrieve the collapsing protohalo regions. We define a protohalo as the collection of particles in the ICs that will all end up in the same halo at $z = 0$. The entire particle set at the ICs can therefore be split into background particles (that will not be part of any virialized structure) and multiple progenitor clumps that will collapse into distinct dark matter haloes.

Due to the fact that CNNs operate upon gridded data, we switch from the spatial particle distribution to a grid-based density contrast by means of the cloud-in-cell algorithm (e.g. Howlett, Manera & Percival 2015). The grid resolution is chosen to be 512^3, so that each cell contains approximately one particle at the ICs. We transform the deposited density contrast

$$\delta = \rho/\bar{\rho} - 1 \tag{1}$$

by unit-variance scaling, where the data are divided by the standard deviation of all pixel values. The distribution of the resulting data set is centred around 0 with a standard deviation of 1. This data pre-processing step has been shown to help in faster convergence of deep learning models (LeCun et al. 1998).

Naylor et al. (2019) presented a novel approach to turn a pixelwise classification problem into a distance-based regression task. Let us assume that an image harbors multiple well-defined objects defined by different labels as seen in Fig. 2 (subplot *b*). All pixels are associated a label that is unique of the enclosing object they belong to, whereas pixels not belonging to any substructure typically carry a zero label. We define the set K_x as the collection of pixels x that belong to a specific object in the image. The residual cells denoted as y define the background of this substructure, that is $y \notin K_x$. The Euclidean distance transformation (EDT) of a multilabelled map B is then defined as $B_d = \mathrm{d}(B)$, where each cell x is assigned the Euclidean distance to the closest background pixel as defined above, that is

$$\mathrm{d}(x) = \min_{y \notin K_x} |x - y|. \tag{2}$$

The transformation is conducted with the EDT[1] package, which implements EDT for multilabel regions. The main problem in binary segmentation is that close or overlapping objects tend to be segmented as one single region. Instead of predicting the protohalo membership of individual pixels in a binary fashion, each cell is assigned the nearest distance to the border of the cluster it resides within.

We therefore construct the distance map of protohalo regions in the following way. Apart from general 6D phase-space information, each dark matter halo in the catalogue also carries the unique ID's of its member particles. Conversely, we record for each particle the mass of the halo it resides in at $z = 0$. If a particle does not belong to a halo it is assigned a halo mass label of 0. This mass information is then traced back for each particle to the ICs at $z = 100$, where it is deposited in the following hierarchical fashion. Each cell of the target map is assigned the largest halo mass label present inside it. If multiple labels are present in a single cell, the deposited value corresponds to the mass of the largest halo, where cells with no halo particles inside are assigned a zero-value corresponding to the background. This top-down approach prioritizes large-scale protohaloes by construction, but since at $z = 100$ approximately one particle resides in each cell, almost all of the halo information is preserved. Moreover, as the labels are directly retrieved from the halo catalogue at $z = 0$, the masses of dark matter haloes and their corresponding protohaloes equal in very good approximation. The entire pipeline is schematically shown in Fig. 2 .

Regressing the distance information as the mapping target offers the major advantage that the networks attention is primarily drawn to the central cluster regions where the density contrast is generally largest. As outlined in Section 3.2, the training loss function compares the predicted and target distance values in a direct way, such that the prediction on central cells is more weighted compared to border pixels. Additionally, as the dark matter haloes mass ranges from 4×10^{12} to $\sim 10^{15}\,\mathrm{M}_{\odot}$ the size of regions, and thus the distance values, vary across different scales. We therefore decide to normalize the distance map of each individual protohalo cluster, such that the value of the central cell equals 1 for all objects. The individually normalized distance map $B_{d,n}$ forms the regression target in the network mapping. We also investigated training on the target of non-normalized distance maps and found that the network primarily focuses on predicting the correct distance value for large-scale objects while often failing on small scales. We show a sample result of these operations in Fig. 2.

2.3 Generating synthetic training samples

We construct the training and test samples from the two deposited particle fields described above. The input i is the unit-variance scaled density contrast whereas the ground target g corresponds to the normalized distance map. The network is trained on regressing the correct distance value from the density contrast, where both fields represent the state at the ICs. The data pairs (i, g) are constructed by a tiling strategy, which is similar to the one presented by Berger & Stein (2019). We divide the entire 512^3 domain into subvolumes of 128^3 where adjacent samples overlap by 64 cells to avoid edge effects. In this manner, the two simulations are each divided into 512 samples, where the innermost 64^3 cells are unique to each subvolume. Each cube can be rotated in six random directions (two per axis), where for each direction exist an additional four possibilities to orient the cube around the given axis, giving a total augmentation factor of 24. We augment the training set by the aforementioned transformations resulting in a total set of 12 288 training samples.

3 NEURAL NETWORK IMPLEMENTATION

CNNs achieve feature extraction from samples by convolving the inputs with filters to produce so-called feature maps. The information

[1] github.com/seung-lab/euclidean-distance-transform-3d

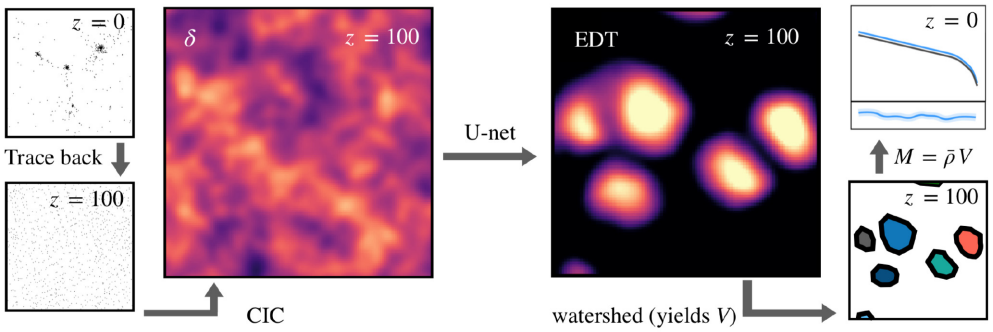

Figure 1. A schematic overview of the entire pipeline. We describe the network mapping and how the training samples of density contrast δ and EDT (defined in equation 2) are constructed in Section 2. We discuss the network architecture, training strategy and results in Section 3. In Section 4, we introduce the watershed algorithm as a key element in the post-processing of the neural network predictions and describe how the halo mass function is reconstructed.

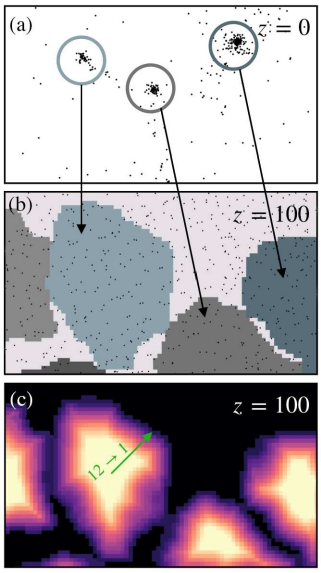

Figure 2. An example situation of the target sample generation pipeline. Particles in haloes at $z = 0$ are traced back to the ICs where the associated halo mass is deposited hierarchically as described in Section 2.2. This results in distinct protohalo regions marked by different colours to better convey individual structures. We then apply the EDT transformation to the entire label image and normalize on an object by object basis to produce the distance map $B_{d,n}$ shown in the third subplot. Also shown in green is the actual and normalized distance value (12 and 1) to the closest background cell for the innermost cell for an example protohalo. Background cells (shown in black) have zero distance. The normalized Euclidean distance information conveys that the target map retains individual objects even though distinct patches might be connected.

stored in these feature maps is subsequently downsampled by pooling operations, which are designed to preserve the important information signals. As data descend deeper into the network, the feature representations generally grow in numbers, while the size of the feature maps decreases due to information pooling. The network architecture is designed in a way that the information of individual feature maps flows together inside deeper convolution layers, allowing the network to learn non-linear combinations of identified features.

3.1 U-net architecture

In this work, we make use of the U-net first introduced by Ronneberger, Fischer & Brox (2015) for solving biomedical image segmentation tasks. The network itself is a fully convolutional autoencoder consisting of two main branches, an encoding and decoding part. As in the general autoencoder case, the information compression is realized by convolutions followed by pooling operations. For image segmentation, the output and input dimensions must match. In order to recover the original input dimensions, the decoding blocks are constructed by a single transpose convolution followed by consecutive convolution operations, which results in upsampled feature maps. A schematic depiction of the entire network is shown in Fig. 3, where the data flow in a U-shape form from the upper left to the lower middle and then reascends along the right-hand side. In this process, the network is able to extract advanced features, but looses localization information at the same time. In order to correct for this loss, Ronneberger et al. (2015) introduced skip connections (often termed *fine-grained feature forwarding* connections), that copy and concatenate the information from the corresponding encoder level with the upflowing data in the decoder part. In this manner, the spatial information from the contraction path is directly transferred to the expanding branch without being passed through the bottleneck and deconvolution operation. Additionally, Ronneberger et al. (2015) also found that the skip connections greatly reduce training time regarding model convergence.

We implement and train our own U-net version in `keras`, a high-level neural networks API (Chollet et al. 2015). The individual convolution blocks are constructed by two subsequent convolutions of filter size ($3 \times 3 \times 3$) and stride 1. The first network layer deploys n_f convolution filters, where we choose $n_f = 12$ in our implementation. After each convolution a non-linear activation function is applied to the data tensors. We choose LReLu, a leaky variation of ReLu (Rectifier-Linear-Unit), with an α-value of 0.05. The main reason for this choice is that deep networks with native ReLu activations have been found to occasionally suffer from vanishing gradients (e.g. Bengio, Simard & Frasconi 1994). The main advantage of LReLu lies in its non-zero gradient, which makes the network more robust as training should in principle never stop. After the second activation in each layer the data are copied and split along two paths. The first path is the skip connection, which concatenates to the upflowing data on the reascending network part. Along the second path, the data flow through a Dropout layer (Srivastava et al. 2014) and finally passes through a single MaxPool operation with filter size ($2 \times 2 \times 2$) and

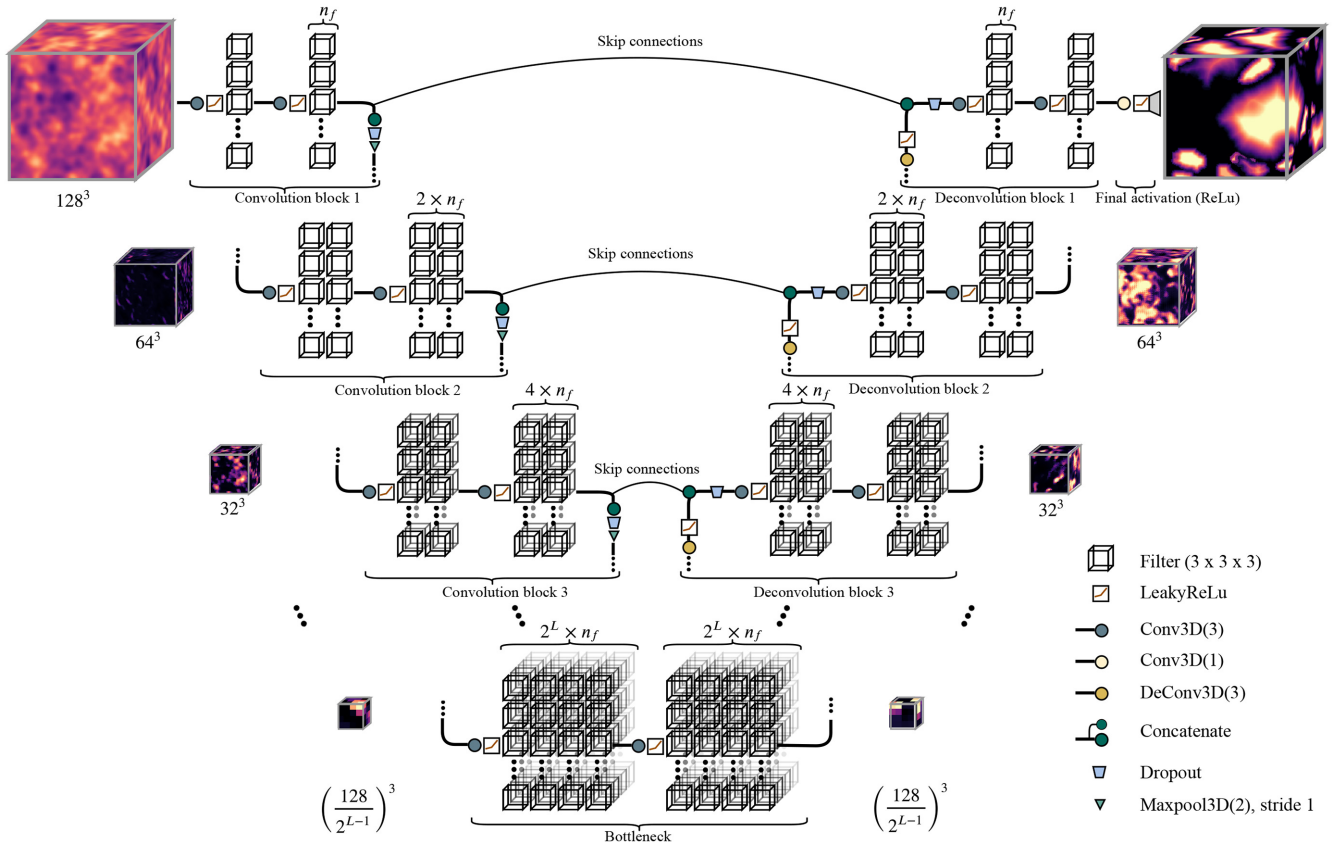

Figure 3. A schematic representation of the fully convolutional U-net implementation used for the regression task. The network is constructed by stacking and connecting convolution blocks down to a bottleneck layer, from which the original input dimensions are reconstructed by upsampling the data through deconvolution blocks. The skip connections concatenate the convolution and deconvolution blocks with same dimension in every corresponding network layer. Also shown are the intermediate layer outputs before each convolution block and after each deconvolution block respectively. With each downward step, the dimensions decrease by a factor of 2 due to the MaxPooling operations, whereas in the decoder part the transpose convolutions upsample the dimension again by the same factor of 2.

Table 1. Summary table of all network hyperparameters and layers for a 5-level version of the U-net with $n_f = 12$ initial filters and filter size of ($3 \times 3 \times 3$) throughout the network. Only `Conv3D` and `Conv3DT` contain trainable parameters.

U-net ($L = 5$, $n_f = 12$) Architecture/operation	Occurrence	
	Number per layer l	Total number
`Conv3D`	$2^{l+1} n_f$	3024
`Conv3DT`	$2^l n_f$	372
`MaxPool`	1 (encoder)	5
`Dropout`	1 (both)	10
`LReLu`	2 (encoder), 3 (decoder)	27
Total free parameters	14.5×10^6	

stride 1. As the data descend the network reaching deeper levels, the architecture of the convolution blocks remains the same with the exception that a total of ($2l \times n_f$) filters are applied, where $l = [1, ..., L]$ is the corresponding layer number. The final network is constructed with a total of five convolution blocks (i.e. $L = 5$). For deeper layers, the network becomes more sensitive to large-scale structures in the input image since the pooling operations cut the corresponding image dimensions in half for each additional layer. As the physical size of one simulation cell is ∼0.2 Mpc, the network downsampling increases the receptive field by a factor of $2^5 = 32$. The deepest network filters of size ($3 \times 3 \times 3$) can therefore learn ∼20 Mpc features in the density field. In the upsampling branch features of all scales are used in conjunction with the spatial information provided by the skip connections to assemble the final prediction. We choose a native ReLu for the final activation function as the target only contains values ≥ 0. The network details are summarized in table 1.

3.2 Training the neural network

The training is conducted on individual batches of size 16 (train-on-batch strategy). For loss function minimization, we use the Adaptive momentum optimizer (Adam) provided in the `keras` (Chollet et al. 2015) package. By the gradient descent algorithm, the network learns relevant relations to accurately predict the distance map from the input density field. However, since a training sample only covers an area of 128^3 cells, one expects the network to be less accurate when predicting the target value for pixels close to the outer border, because important parts of the density field are stored in the next adjacent sub-box. Hence, the approach of splitting the entire simulation domain into sub-boxes introduces an obstacle in the training algorithm caused by edge effects. To correct for this, we implement a custom loss function based on the normalized L_1 loss (normalized mean-absolute error), that is only sensitive to pixels inside a pre-defined receptive field. The buffer size is chosen to be 16, so that only the innermost 96^3 cube is considered when evaluating the cost function.

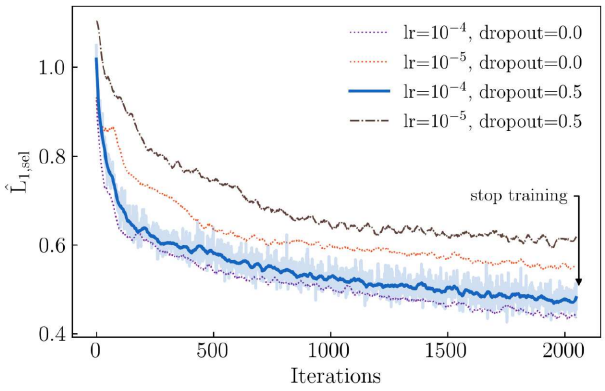

Figure 4. The evolution of the loss function $\hat{L}_{1,\mathrm{sel}}$ over the entire ∼2000 training iterations, after which the loss improvements become marginally small and training is stopped. The dropout rate is set to 0.5 throughout the entire training process for the fiducial run (shown in blue). We also show the loss evolution for additional runs with varied learning rate and dropout values.

We choose a smaller buffer size compared to the buffer of the individual sub-boxes, because otherwise larger structures would not entirely fit inside the receptive field of the loss function and their identification would have to span across neighbouring sub-boxes. This is undesired, since the neural network is only presented with a single instance during prediction time. Formally, the selective loss function is constructed as

$$\hat{L}_{1,\mathrm{sel}} = \frac{1}{n\bar{g}} \sum_{k}^{n} (g_k - p_k)^2 \Big|_{\square 96^3} \tag{3}$$

where g_k and p_k denote individual cells of ground truth and prediction. We choose to normalize the loss function by dividing by the mean ground truth distance $\bar{g}$, since different sub-boxes show largely varying distance values depending on how many collapsing regions are located inside it. The dropout rate is set to 0.5 during the entire training period for the fiducial run.

In Fig. 4, we show the evolution of the loss function, where after ∼2000 iterations loss minimization stagnates and training is stopped. The entire training process took 24 h on a single Tesla K-80 GPU card. The network reaches a loss value of ∼0.6 in a relatively short training period. The subsequent gradient descent however is slow resulting in the characteristic exponential tail that is often encountered when training deep learning algorithms. The small training set of only 512 unique samples presents another challenge as the global minimum may be surrounded by local minima that are characteristic for the training set. Moreover, the comparatively small batch size of 16 as well as the train-on-batch strategy make the gradient descent less smooth. Even though in principle this can be overcome by changing the training strategy, the shortcoming is the loss of quick access control to the training algorithm because the network weights do not get updated as frequently as in the train-on-batch strategy. For comparison we show in Fig. 4, the loss evolution of additional runs where we slightly change the learning and dropout rates. Independent of the learning rate, a higher dropout rate results in larger loss values throughout the training as only a fraction of the network is active during a training iteration. Since our training set is comparatively small, we prefer the additional benefit of network regularization over a slightly lower loss value. Choosing smaller learning rates generally results in slower gradient descent and thus longer training times. We find that setting the learning rate to 10^{-4} consistently outperformed networks with lower learning rates. Additionally, any training runs with larger learning rates ($\sim 10^{-3}$) failed to converge and we therefore empirically use the training run shown in solid blue (learning rate $= 10^{-4}$, dropout $= 0.5$) for our further analysis.

The trained network is then used to predict the distance map for the test simulation. A slice of the test box is displayed in Fig. 5. In general, the network manages to identify the important regions of gravitational collapse and to estimate the corresponding protohalo sizes. For large and intermediate scales, the general sizes of connected structures in the distance map agree well with the ground truth. There are however individual cases where the compact and intermingled structure of many nearby clusters poses a great challenge for the network as it is not able to accurately split the individual regions in the correct way. Occasionally, two intermediately sized protohalo regions will be merged by the network to produce a single cluster and vice versa. We show an example of such a faulty network prediction in the zoomed-in subplot A in Fig. 5, where the network does infact predict the right size of the collapsing regions but splits the density patch into two sub protohaloes. Accurately predicting the distances of small-scale structure poses the largest challenge for the network as there are at least three possible interplaying reasons for this behaviour:

(i) The signal strength in the density contrast is often too low for the network to retrieve any valuable collapse dynamics on small scales.

(ii) The corresponding region in the input density field does in fact show signs of an overdensity but through gravitational dynamics the small patch is either dispersed or merges with a larger nearby cluster and its isolated signal is lost. This is an expected behaviour in hierarchical cosmologies as larger structures grow by merging events of smaller haloes. We find that the network is not able to accurately predict the outcome of complicated gravitational dynamics for small-scale structures.

(iii) Regions that will form haloes at $z = 0$ with masses slightly below the halo mass barrier of $4 \times 10^{12}\,\mathrm{M}_{\odot}$ will not appear in the ground truth distance map, even though the signal is present in the density field at the ICs. The same effect can also happen in the opposite direction, where the network discards a patch as a potential protohalo even though this patch might accumulate enough mass over time to exceed the mass threshold. This is a natural shortcoming of imposing a hard threshold on small-scale structure.

4 FROM DISTANCE INFORMATION TO SEGMENTATION MAPS

In the following section, we describe how the output from the neural network is post-processed to predict the mass of individual protohalo regions and to construct the final halo mass function as summarized in Fig. 1.

Having trained the network to predict the desired distance information from the density contrast, we construct an adaptive algorithm that retrieves the boundaries of individual protohalo regions as defined by the metric in equation (2). We make use of the watershed algorithm, a fast and reliable tool used in digital image processing for segmenting data with overlapping regions. The name refers metaphorically to a geological watershed, which separates adjacent drainage basins, as it treats the image it operates upon like a topographic map. With the brightness of each pixel representing its height (or distance as defined in equation 2), the algorithm finds the lines that run along the tops of ridges and is able to separate

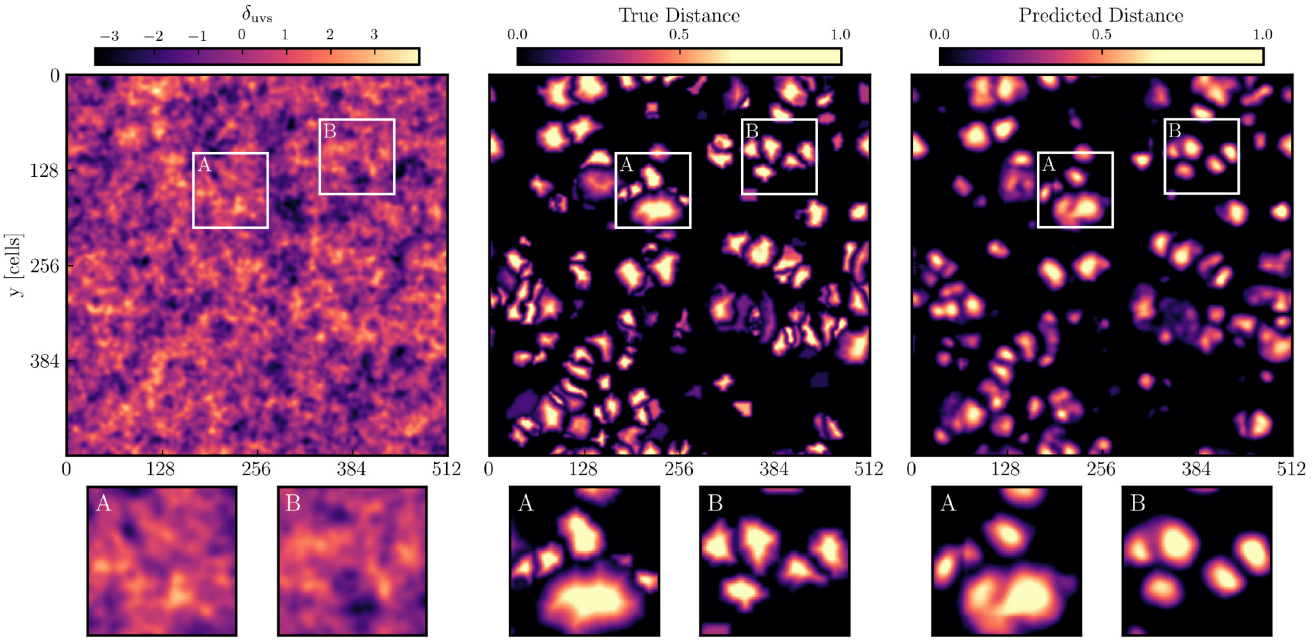

Figure 5. An entire slice of the test simulation box displaying the input density contrast at $z = 100$ (left), raw and unprocessed prediction of the neural network (right) and ground truth (middle) of the normalized distance map. Also shown are insets of two regions conveying the performance of the neural network in more detail. In case B, the network manages to predict the exact number of distinct clusters as well as a good estimate on the corresponding sizes, whereas case A shows a problematic prediction regarding protohalo identification where a major density patch is split into two substructures. This behaviour is unwanted as the reconstructed region will suffer from oversegmentation.

adjacent regions from one another. We suggest that the interested reader consult Kornilov & Safonov (2018) for a detailed overview.

We adopt the marker-based version of the watershed algorithm, which is extremely useful for this purpose as the distance map is the only needed ingredient. In a preliminary step, the algorithm runs a local-maxima finder to identify the positions of distance peaks, which play the important role of preselected markers. In a subsequent step, sources are placed at the marker positions, from which the image is flooded until water basins attributed to different markers meet on watershed lines. The resulting set of segmented regions constitutes a watershed segmentation by flooding.

4.1 Generating halo catalogues and bias correction

The native watershed algorithm however has a natural shortcoming when reconstructing images of structures with different scales. An example is shown in Fig. 6, where two clusters of different sizes are very close to one another. The middle pane shows the reconstruction from the native watershed algorithm. The general problem is that by simply flooding the image, any region where pixels have non-zero values is filled and gets assigned to one of the two clusters, which generally results in overestimated sizes when compared to the ground truth protohaloes. This behaviour introduces a bias that changes with mass scale as it originates from the imperfect network predictions at cluster borders.

We correct for this overestimation by introducing an adaptive element to the native watershed output that restricts the regions, within which the algorithm can fill the image. This is achieved by thresholding the distance map of each individual cluster with an adaptive value that is found by calibration with the ground truth halo mass function in the following strategy. We switch back to the unnormalized cell-based distance d by means of a spherical

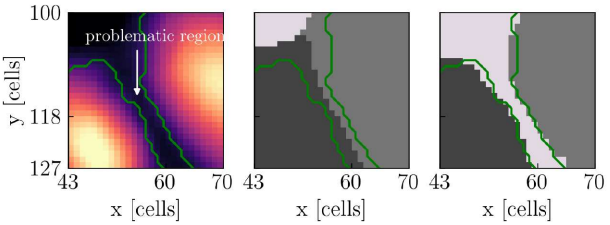

Figure 6. An example protohalo region that conveys very well the need for an adaptive version of the watershed algorithm. On the left, a zoomed-in slice of two predicted clusters is shown. The middle pane shows the raw cluster reconstruction from the native watershed algorithm, whereas the result of the adaptive post-processed version is shown on the right. Given the true protohalo boundaries overlaid in green in all three plots, it is evident that the adaptive watershed algorithm yields better estimates regarding the individual sizes of neighbouring clusters, as problematic regions are filtered away.

approximation for each cluster,

$$d = \hat{d} \times \frac{3}{4\pi} V^{1/3}, \tag{4}$$

where V denotes the uncorrected size in terms of individual cells and $\hat{d}$ is the normalized distance value from the prediction. By thresholding the distance map of a cluster with increasing values, we trace the evolution of the mass and the averaged distance values inside it. This results in unique trajectories in the $(V, \bar{d})$-space for every protohalo region (see Fig. 7). A rapid change in mass, while the mean distance remains constant is an indication for a smeared border that is not well defined. The information of all trajectories is accumulated in a 2D histogram that represents the occurrences of intersections in each $(V, \bar{d})$-segment. The ground truth halo mass function yields the information of how many haloes are situated in a given size bin. Through direct calibration we then identify for each

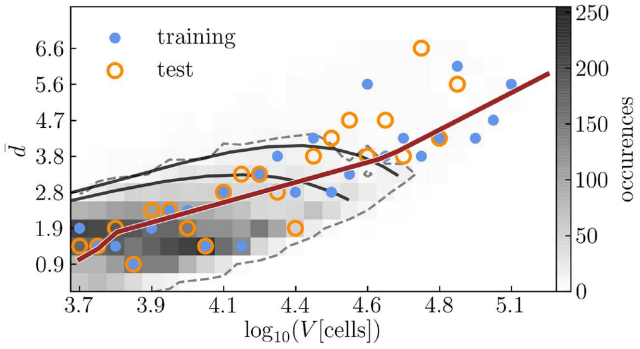

Figure 7. We show two example trajectories in the $(V, \bar{d})$-space of two individual haloes in black, where increasing the contour threshold results in smaller and thus less massive regions. Beyond the dashed contour individual bins contain less than 10 intersections, implying very low statistics. Calibrating with the ground truth halo mass function yields the scatter markers where blue dots and orange circles represent training and test data, where it is apparent that the overestimation follows the same trend in both data sets. We manually fit a piecewise linear function (red) through the training scatter to retrieve the adaptive relation between V and $\bar{d}$. The corrected protohalo sizes then correspond to the bin where their trajectories intersect the fitted relation.

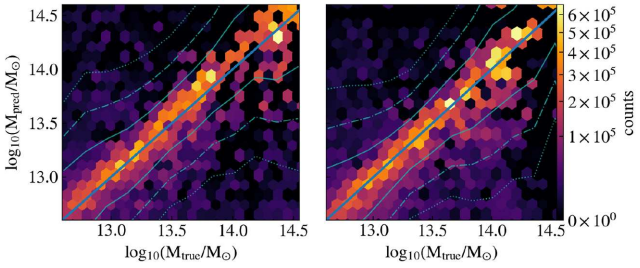

Figure 8. We show predicted against true voxel masses for the training (left) and test (right) simulation for voxels above the pre-defined threshold of $4 \times 10^{12}\,\mathrm{M}_\odot$ where the blue line indicates the case of perfect prediction. The bluegreen lines indicate the 1σ (solid), 2σ (dashed–dotted), and 3σ (dotted) contour in each fixed mass bin.

mass scale, the distance bin that offers the smallest residual in terms of halo number with respect to the ground truth. The identified bins (shown as blue data points in Fig. 7) are then fitted with a piecewise linear function to understand the bias introduced by watershed as a corrective relation between V and $\bar{d}$. We are constricted to choose a comparatively low number of bins, as the small sample number of ∼1500 haloes does not offer enough statistics and is unreliable for finer binning.

We implement the entire adaptive element as a post-processing step following the native watershed flooding, meaning that the entire cluster size retrieval includes the following two steps. First, the segmented image is reconstructed by the native watershed algorithm where sizes are generally overestimated. In a second step, the different reconstructed regions are thresholded depending on the mass bin where their corresponding trajectories intersect the piecewise linear fit. The numerical reason behind this approach is that the correction remains minimal because the actual masses are held as large as possible since only the outermost borders are cut away. In this manner we construct an algorithm that is sensitive to different cluster scales as small- and intermediate-scale structures are prevented from being merged with nearby clusters. We find that this adaptive thresholding algorithm recovers the individual sizes much better compared to a uniform threshold value, which would suffice in the case of the neural network achieving close to perfect precision. An example of this correction is showed in Fig. 6. We emphasize that this step is purely due to the imperfections of the network predictions. Training deeper models with larger training sets at higher resolution may eradicate the need of this adaptive post-processing element entirely.

The collection of reconstructed protohalo sizes is then converted to the final mass catalogue by means of the background density at the ICs, that is $M = \bar{\rho} V$. This strategy works very well in the linear regime ($\delta \approx 0$). In fact, as the constructed grid matches the particle resolution of 512^3, there is approximately one particle in each grid cell, such that final halo masses at $z = 0$ equal the protohalo masses in very good agreement. Thus, predicting the protohalo masses allows to reconstruct the halo mass function of dark matter haloes at $z = 0$. In Fig. 9, we show the true and predicted cumulative halo mass functions as well as the percentage deviation from the ground truth for the training and test simulation. The reconstructed statistics match the ground truth catalogues for most mass bins within an uncertainty of 10 per cent. The deviation is largest for very massive protohalo regions of the order $\sim 10^{14}\,\mathrm{M}_\odot$ and beyond. Given that the total numbers of these large-scale objects in the entire simulation box is comparatively low, the variety of cases that the neural network is presented with during training is especially limited on these scales. We retrieve a total of 1484 protohalo regions for the training simulation and 1541 thereof in the test case (5 per cent deviation in both cases). The neural network as well as the piecewise linear correction fit generalize well to unseen data samples from the test set (Figs 5 and 7).

From tile A in Fig. 5 we see that in some cases the network predicts multiple protohalo patches when the true configuration simply shows one massive structure (oversegmentation). Occasionally, the opposite behaviour of unwanted merging scenarios happens as well (undersegmentation). For this reason, a direct one-to-one halo comparison in the predicted and true field is difficult as one would have to define criterions depending on the amount of overlap between a predicted and true patch to identify them as the same. Instead, we show a mass comparison on a voxel-by-voxel basis to understand the predictive power of the algorithm as well as the scatter in the relation of prediction versus truth in Fig. 8. We see that the general trend closely follows the perfect case (indicated by the blue line) with a comparatively small amount of scatter that increases for larger halo masses. This is an expected behaviour as the number of haloes decreases for larger mass bins (see e.g. Fig. 9).

5 CONCLUSIONS AND FUTURE WORK

We have presented a framework for the task of identifying protohalo regions on a one-by-one basis directly from the ICs of N-body simulations. We designed this challenging task as a hybrid approach consisting of pure Deep Learning (U-net) paired with an image reconstruction technique (watershed algorithm). We showed that by reformulating the underlying classification task as a distance-based regression problem, the comparatively small network with $\sim 14.5 \times 10^6$ trainable parameters is in fact able to retrieve collapsing regions despite being trained on data samples from only one simulation box.

We argued that the native marker-based watershed algorithm is not sensitive enough for reconstructing the exact halo masses as it generally overpredicts the sizes of individual regions due to blindly flooding the 3D image. We implemented a computationally fast addition to complement the native algorithm with an adaptive post-processing element that is calibrated with the training simulation.

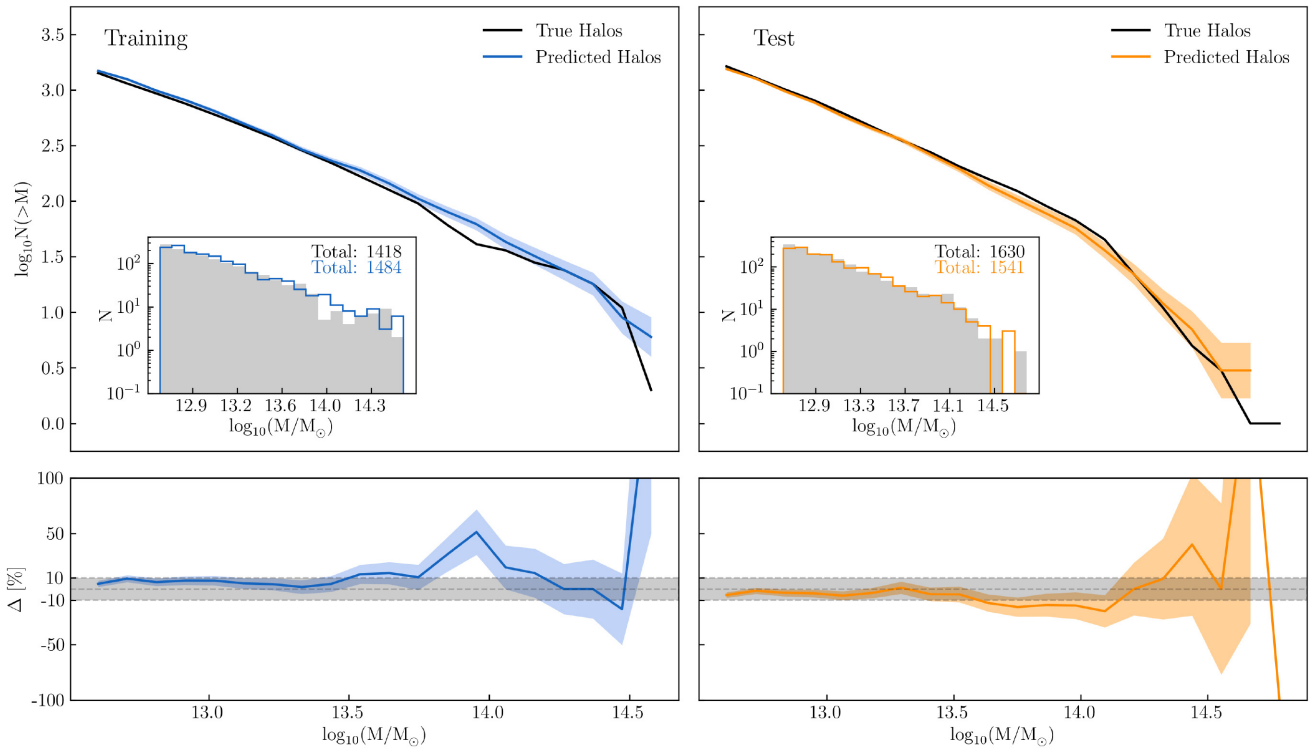

Figure 9. Reconstructed cumulative halo mass functions for the training (left) and test (right) simulations at $z = 0$, where $N(>M)$ is the number of haloes with a mass greater than M. The corresponding ground truths are shown in black and the colour shaded regions denote the Poisson uncertainty in the corresponding mass bin. Also shown as insets are the differential versions with the predicted and true halo number counts as well as the percentage deviation from the ground truth halo mass functions.

We find that this hybrid model is able to reconstruct the halo mass function at $z = 0$ within an uncertainty of $\sim$10 per cent for most mass bins. In addition to the statistical precision, the model also achieves good agreement when comparing the actual sizes and positions of protohalo regions on a one-by-one basis as seen in Fig. 5. The hybrid model is designed to be robust against many complications that arise from fully non-linear dynamics in *N*-body simulations. The pixelwise Euclidean distance map as the target is completely free from any assumptions on the cluster morphology (e.g. spherical approximation), and bypasses the difficulties of training a multilabel classification task, where each protohalo region would be treated as a different class. Generally, the network achieves good precision on the regressed distance information and is able to isolate distinct protohalo regions.

However, a weakness of the current pipeline is that it occasionally fails to predict halo formation if multiple density patches ending up in a single halo originate from disconnected regions (as seen in case A in Fig. 5). It is possible that these faulty predictions originate from cases where the density configuration alone might not yield enough information regarding how patches merge or split. For this problem, we propose to include the velocity field as an additional information carrier when training the neural network as it could provide missing information on the fragmentation of individual density patches. The current setup can easily be extended to predict more halo properties since this technique only depends on how the target fields are constructed numerically. Furthermore, the current pipeline was constructed by imposing a hard threshold on small-scale haloes. Compared to the simulation this is a shortcoming of our pipeline, as the network is not able to probe the halo mass function at very small scales. As briefly mentioned before, the reason for this choice is that small-scale patches clustered around larger structures make the protohalo borders less well resolved, often resulting in gaps between patches that are only a couple grid pixels wide. In our experiments, we found that the identification power of the network relies on the prerequisite that gaps between patches are resolved well enough. This aspect can in principle be overcome by choosing a higher grid resolution, while keeping the mass resolution the same. However, in such a scenario one would have to adapt the deposition strategy as it depends on the fact that the number of dark matter particles and the number of grid cells be the same, such that each grid cell at the ICs includes approximately one particle.

Future installments will investigate to what extent an improved hybrid model can predict merger scenarios based on information about the locations of overdensities and their relative velocities and gradients thereof to extend the probing range to even smaller scales. We expect that training deeper networks on larger data sets with higher resolution will decrease the systematic errors on the halo mass function. In our training strategy, we made use of the normalized L_1 loss function, which measures deviations on a pixel-by-pixel level. Formulating a loss function that is more receptive and physically motivated regarding the underlying task of protohalo identification could offer a large opportunity for directing the learning process more towards the physics behind collapse dynamics (e.g. Karpatne et al. 2017).

The problem of fully non-linear structure formation across a wide range of scales offers a challenging opportunity to test and fine-tune different machine learning approaches. From the results of this work, we conclude that Deep Learning models are capable of learning the relevant combinations from the input signals to identify collapsing regions in full *N*-body simulations. In this sense, the presented

hybrid model offers a powerful base framework towards designing more precise algorithms with the eventual objective to build cosmic emulators for fast sampling of dark matter halo catalogues.

ACKNOWLEDGEMENTS

RF acknowledges financial support from the Swiss National Science Foundation (grant no. 157591). We thank the anonymous referee for his valuable input that helped improving this work.

DATA AVAILABILITY STATEMENT

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Agarwal S., Davé R., Bassett B., 2017, MNRAS, 478, 3410
Aragon-Calvo M. A., 2019, MNRAS, 484, 5771
Bengio Y., Simard P., Frasconi P., 1994, IEEE Trans. Neural Netw., 5, 157
Berger P., Stein G., 2019, MNRAS, 482, 2861
Bond J. R., Cole S., Efstathiou G., Kaiser N., 1991, ApJ, 379, 440
Charnock T., Lavaux G., Wandelt B. D., Boruah S. S., Jasche J., Hudson M. J., 2020, MNRAS, 494, 50
Chollet F. et al., 2015, Keras. Available at: https://keras.io
Eisenstein D., Hut P., 1998, ApJ, 498, 137
Feldmann R., Faucher-Giguère C.-A., Kereš D., 2019, ApJ, 871, L21
Guo Q., White S., Li C., Boylan-Kolchin M., 2010, MNRAS, 404, 1111
He S., Li Y., Feng Y., Ho S., Ravanbakhsh S., Chen W., Póczos B., 2019, Proc. Natl. Acad. Sci., 116, 13825
Howlett C., Manera M., Percival W. J., 2015, Astron. Comput., 12, 109
Isensee F., Kickingereder P., Wick W., Bendszus M., Maier-Hein K. H., 2018, Brain tumor segmentation and radiomics survival prediction: contribution to the brats 2017 challenge, preprint (arXiv:1802.10508)
Karpatne A., Watkins W., Read J., Kumar V., 2017, Physics-guided neural networks (pgnn): an application in lake temperature modeling, preprint (arXiv:1710.11431)
Kodi Ramanah D., Charnock T., Lavaux G., 2019, Phys. Rev. D, 100, 043515
Kornilov A., Safonov I., 2018, J. Imag., 4, 123
Krizhevsky A., , Sutskever I., Hinton G. E., 2012, in Advances in Neural Information Processing Systems, Vol. 25. p. 1097
LeCun Y., Bottou L., Orr G. B., Müller K.-R., 1998, in Neural Networks: Tricks of the Trade.
Liu W., Rabinovich A., Berg A. C., 2015, Parsenet: looking wider to see better, preprint (arXiv:1506.04579)
Lucie-Smith L., Peiris H. V., Pontzen A., Lochner M., 2018, MNRAS, 479, 3405
Lucie-Smith L., Peiris H. V., Pontzen A., 2019, MNRAS, 490, 331
Mathuriya A. et al., 2018, preprint (arXiv:1808.04728)
Milletari F., Navab N., Ahmadi S.-A., 2016, V-net: fully convolutional neural networks for volumetric medical image segmentation, preprint (arXiv:1606.04797)
Naylor P., Laé M., Reyal F., Walter T., 2019, IEEE Trans. Med. Imag., 38, 448
Noh H., Hong S., Han B., 2015, Learning deconvolution network for semantic segmentation. preprint (arXiv:1505.04366)
Ntampaka M., Eisenstein D. J., Yuan S., Garrison L. H., 2020, ApJ, 889, 151
Press W. H., Schechter P., 1974, ApJ, 187, 425
Reed D., Gardner J., Quinn T., Stadel J., Fardal M., Lake G., Governato F., 2003, MNRAS, 346, 565
Ronneberger O., Fischer P., Brox T., 2015, U-net: convolutional networks for biomedical image segmentation, preprint (arXiv:1505.04597)
Sheth R. K., Mo H. J., Tormen G., 2001, MNRAS, 323, 1
Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res., 15, 1929
Stadel J. et al., 2000
Wechsler R. H., Tinker J. L., 2018, ARA&A, 56, 435
Zhang X., Wang Y., Zhang W., Sun Y., He S., Contardo G., Villaescusa-Navarro F., Ho S., 2019, From dark matter to galaxies with convolutional networks, preprint (arXiv:1902.05965)

This paper has been typeset from a TEX/LATEX file prepared by the author.